

RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFF	FFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFF	FFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFF	FFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFF	FFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFF	FFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFF	FFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	

```

RRRRRRRR      SSSSSSSS  DDDDDDDD  EEEEEEEEE  RRRRRRRR  MM      MM
RRRRRRRR      SSSSSSSS  DDDDDDDD  EEEEEEEEE  RRRRRRRR  MM      MM
RR      RR  SS      DD      DD  EE      RR      MMMM  MMMM
RR      RR  SS      DD      DD  EE      RR      MMMM  MMMM
RR      RR  SS      DD      DD  EE      RR      MM  MM  MM
RRRRRRRR      SSSSSS      DD      DD  EEEEEEE  RRRRRRRR  MM      MM
RRRRRRRR      SSSSSS      DD      DD  EEEEEEE  RRRRRRRR  MM      MM
RR      RR      SS      DD      DD  EE      RR      MM      MM
RR      RR      SS      DD      DD  EE      RR      MM      MM
RR      RR      SS      DD      DD  EE      RR      MM      MM
RR      RR      SSSSSSSS  DDDDDDDD  EEEEEEEEE  RR      RR  MM      MM
RR      RR      SSSSSSSS  DDDDDDDD  EEEEEEEEE  RR      RR  MM      MM

```

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

```

....
....
....
....

```

```
1 0001 0 XTITLE 'Misc. error checks and messages.'
2 0002 0 MODULE RSDERM (
3 0003 0 IDENT = 'V04-000'
4 P 0004 0 XBLISS32[
5 P 0005 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)
6 0006 0 ]
7 0007 0 ) =
8 0008 1 BEGIN
9 0009 1
10 0010 1 *****
11 0011 1 *
12 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14 0014 1 * ALL RIGHTS RESERVED.
15 0015 1 *
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 * TRANSFERRED.
22 0022 1 *
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 * CORPORATION.
26 0026 1 *
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *****
31 0031 1
32 0032 1 ++
33 0033 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
34 0034 1
35 0035 1 ABSTRACT: Residual error messages. Issues error messages for various
36 0036 1 error conditions for which no error messages have been
37 0037 1 issued.
38 0038 1
39 0039 1 Also contains error detection and error message routines
40 0040 1 that are called from more than one location.
41 0041 1
42 0042 1 ENVIRONMENT: Transportable
43 0043 1
44 0044 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978
45 0045 1
46 0046 1
```

RSDERM
V04-000

Misc. error checks and messages.
Revision History

F 11
16-Sep-1984 01:40:15
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]RSDERM.BLI;1

Page 2
(2)

```

: 48
: 49
: 50
: 51
: 52
: 53
: 54
: 55
: 56
: 57
: 58
: 59
: 60
: 61
: 62
: 63
: 64

0047 1 %SBTTL 'Revision History'
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1
0052 1 006 REM00006 Ray Marshall 22-June-1983
0053 1 Modified routine TSTTFE to now look to see if we are within a
0054 1 .LITERAL. If so, it issues an error and closes forces a
0055 1 .END LITERAL.
0056 1
0057 1 005 KFA00005 Ken Alden 18-Mar-1983
0058 1 Made SAVE/RESTORE related items visible to DSR.
0059 1
0060 1 004 REM00004 Ray Marshall 07-Mar-1983
0061 1 Global edit of all modules. Updated module names, idents,
0062 1 copyright dates. Changed require files to BLISS library.
0063 1
0064 1 !--
```

```

66 0064 1 %SBTTL 'Module Level Declarations'
67 0065 1
68 0066 1 REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
69 0197 1
70 0198 1
71 0199 1 !! TABLE OF CONTENTS:
72 0200 1 !!
73 0201 1 FORWARD ROUTINE
74 0202 1     tstblk : NOVALUE,
75 0203 1     tstcnd : NOVALUE,
76 0204 1     tstres : NOVALUE,
77 0205 1     tsttfe : NOVALUE,
78 0206 1     pntbac : NOVALUE,
79 0207 1     negind : NOVALUE,
80 0208 1     remneg : NOVALUE,
81 0209 1     xmarg  : NOVALUE,
82 0210 1     remmrg : NOVALUE;
83 0211 1
84 0212 1 !!
85 0213 1 !! INCLUDE FILES:
86 0214 1 !!
87 0215 1
88 0216 1 LIBRARY 'NXPORT:XPORT';         ! XPORT Library
89 0217 1
90 U 0218 1 %IF DSRPLUS %THEN
91 U 0219 1 LIBRARY 'REQ:DPLLIB';         ! DSRPLUS BLISS Library
92 0220 1 %ELSE
93 0221 1 LIBRARY 'REQ:DSRLIB';         ! DSR BLISS Library
94 0222 1 %FI
95 0223 1
96 0224 1
97 0225 1 !! MACROS:
98 0226 1 !!
99 0227 1
100 0228 1 EQUATED SYMBOLS:
101 0229 1
102 0230 1
103 0231 1 OWN STORAGE:
104 0232 1
105 0233 1
106 0234 1 !! EXTERNAL REFERENCES:
107 0235 1 !!
108 0236 1
109 0237 1 EXTERNAL
110 0238 1     gca : gca_definition,
111 0239 1     ifstk : ifstack,
112 0240 1     savstk : savstack,
113 0241 1     irac  : irac_definition;
114 0242 1
115 0243 1 EXTERNAL
116 0244 1     frmstd,
117 0245 1     frmstk : form_stack;
118 0246 1
119 0247 1 EXTERNAL LITERAL             !Error messages
120 0248 1     rnfak,
121 0249 1     rnflde,
122 0250 1     rnfmei,

```

RSDERM
V04-000

Misc. error checks and messages.
Module Level Declarations

H 11
16-Sep-1984 01:40:15
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]RSDERM.BLI;1

Page 4
(3)

123	0251	1	rnfmrc,
124	0252	1	rnfmrg,
125	0253	1	rnfmrs,
126	0254	1	rnfnia,
127	0255	1	rnfnic,
128	0256	1	rnfskc,
129	0257	1	rnftfe;
130	0258	1	
131	0259	1	EXTERNAL ROUTINE
132	0260	1	erm,
133	0261	1	ermf,
134	0262	1	ermn,
135	0263	1	lit,
136	0264	1	savres,
137	0265	1	stkfrm;
138	0266	1	
139	0267	1	

RS
V0

```

141 0268 1 %SBTTL 'TSTBLK -- test for unclosed .NOTE or .LIST directives'
142 0269 1 GLOBAL ROUTINE tstblk (depth) : NOVALUE =
143 0270 1
144 0271 1 ++
145 0272 1 FUNCTIONAL DESCRIPTION
146 0273 1
147 0274 1 This routine checks to see if FRMSTK is ok: i.e., certain open
148 0275 1 .NOTE and .LIST commands have been closed. If there are any
149 0276 1 unclosed commands it issues an error message and pops the stack.
150 0277 1
151 0278 1 FORMAL PARAMETERS:
152 0279 1
153 0280 1 DEPTH specifies the maximum allowed depth of .REQUIRE commands.
154 0281 1
155 0282 1 IMPLICIT INPUTS: None
156 0283 1
157 0284 1 IMPLICIT OUTPUTS: None
158 0285 1
159 0286 1 ROUTINE VALUE:
160 0287 1 COMPLETION CODES: None
161 0288 1
162 0289 1 SIDE EFFECTS: None
163 0290 1
164 0291 1 --
165 0292 1
166 0293 2 BEGIN
167 0294 2
168 0295 2 IF .frmstd EQL 0 THEN
169 0296 2 RETURN; !No unclosed lists, notes, or literals.
170 0297 2
171 0298 2 IF .depth GTR .frmstk [.frmstd, frmstk_req_d] THEN
172 0299 2 RETURN; !Unclosed lists or notes, but in files still open.
173 0300 2
174 0301 2 erm (rnftfe, 0, 0); !There are unclosed lists/notes to be reported.
175 0302 2 !Now point the user back to those commands that have not been closed
176 0303 2
177 0304 2 WHILE ((.frmstd GTR 0) AND (.depth LEQ .frmstk [.frmstd, frmstk_req_d])) DO
178 0305 3 BEGIN
179 0306 3 pntbac ();
180 0307 3 stkfrm (-1); !Unstack one entry, regardless of identity.
181 0308 3 END;
182 0309 2
183 0310 1 END; !End of TSTBLK

```

.TITLE RSDEM Misc. error checks and messages.
.IDENT \V04-000\

.EXTRN GCA, IFSTK, SAVSTK
.EXTRN IRA, FRMSTD, FRMSTK
.EXTRN RNFBK, RNFLDE, RNFMEL
.EXTRN RNFMRC, RNFMRG, RNFMRS
.EXTRN RNFNIA, RNFNIC, RNFSKC
.EXTRN RNFTFE, ERM, ERML
.EXTRN ERMN, LIT, SAVRES
.EXTRN STKFRM

						.PSECT \$CODE\$,NOWRT,2		
		000C	00000	.ENTRY	TSTBLK, Save R2,R3		: 0269	
53	00000000G	EF	9E 00002	MOVAB	FRMSTD, R3			
52	00000000G	EF	9E 00009	MOVAB	FRMSTK-20, R2			
50		63	D0 00010	MOVL	FRMSTD, R0		: 0295	
		3B	13 00013	BEQL	2\$			
50		OF	C4 00015	MULL2	#15, R0		: 0298	
6240	04	AC	D1 00018	CMPL	DEPTH, FRMSTK-20[R0]			
		31	14 0001D	BGTR	2\$			
		7E	7C 0001F	CLRQ	-(SP)		: 0301	
	00000000G	8F	DD 00021	PUSHL	#RNFTFE			
00000000G	EF	03	FB 00027	CALLS	#3, ERM			
50		63	D0 0002E 1\$:	MOVL	FRMSTD, R0		: 0304	
		1D	15 00031	BLEQ	2\$			
50		OF	C4 00033	MULL2	#15, R0			
6240	04	AC	D1 00036	CMPL	DEPTH, FRMSTK-20[R0]			
		13	14 0003B	BGTR	2\$			
00000000V	EF	00	FB 0003D	CALLS	#0, PNTBAC		: 0306	
	7E	01	CE 00044	MNEGL	#1, -(SP)		: 0307	
00000000G	EF	01	FB 00047	CALLS	#1, STKFRM			
		DE	11 0004E	BRB	1\$: 0304	
		04	00050 2\$:	RET			: 0310	

; Routine Size: 81 bytes, Routine Base: \$CODE\$ + 0000

```
185 0311 1 %SBTTL 'TSTCND -- Check and report missing .ENDIFs'
186 0312 1 GLOBAL ROUTINE tstcnd (depth) : NOVALUE =
187 0313 1
188 0314 1 ++
189 0315 1 FUNCTIONAL DESCRIPTION
190 0316 1
191 0317 1
192 0318 1 FORMAL PARAMETERS:
193 0319 1
194 0320 1 DEPTH specifies the maximum allowed depth of .REQUIRE commands.
195 0321 1
196 0322 1 IMPLICIT INPUTS: None
197 0323 1
198 0324 1 IMPLICIT OUTPUTS: None
199 0325 1
200 0326 1 ROUTINE VALUE:
201 0327 1 COMPLETION CODES: None
202 0328 1
203 0329 1 SIDE EFFECTS: None
204 0330 1
205 0331 1 --
206 0332 1
207 0333 2 BEGIN
208 0334 2 !Check for missing .ENDIF commands. This is temporary coding. In a
209 0335 2 !later version this code should point back to the opening commands,
210 0336 2 !like the code above does for .LIST and .NOTE commands.
211 0337 2
212 0338 2 IF .ifstk [0, ifstk_depth] EQL 0 THEN
213 0339 2 RETURN; !There are no missing .ENDIF commands.
214 0340 2
215 0341 2 IF .depth EQL 0 THEN
216 0342 2 erml (rnfmei) ! Missing .ENDIF commands, and end of file.
217 0343 2 ELSE
218 0344 2 IF .ifstk [.ifstk [0, ifstk_depth], ifstk_req_d] GEQ .depth THEN
219 0345 2 erml (rnfskc); ! Missing .ENDIF commands detected when
220 0346 2 ! unstacking .REQUIRE files
221 0347 2
222 0348 1 END; !End of TSTCND
```

		0000 00000	.ENTRY	TSTCND, Save nothing	: 0312
50	00000000G	EF D0 00002	MOVL	IFSTK, R0	: 0338
		2A 13 00009	BEQL	3\$	
	04	AC D5 0000B	TSTL	DEPTH	: 0341
		08 12 0000E	BNEQ	1\$	
	00000000G	8F DD 00010	PUSHL	#RNFMEI	: 0342
		16 11 00016	BRB	2\$	
50		20 C4 00018 1\$:	MULL2	#32, R0	: 0344
	00000000GEF	40 9F 0001B	PUSHAB	IFSTK+16[R0]	
04	AC	9E D1 00022	CMPL	@(SP)+, DEPTH	
		0D 19 00026	BLSS	3\$	
	00000000G	8F DD 00028	PUSHL	#RNFSKC	: 0345
00000000G	EF	01 FB 0002E 2\$:	CALLS	#1, ERML	
		04 00035 3\$:	RET		: 0348

RSDERM
V04-000

Misc. error checks and messages,
TSTCND -- Check and report missing .ENDIFs

L 11
16-Sep-1984 01:40:15
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]RSDERM.BLI;1

Page 8
(5)

; Routine Size: 54 bytes, Routine Base: \$CODE\$ + 0051

```

224 0349 1 %SBTTL 'TSTRES -- Check for, report, and restore missing .RESTOREs'
225 0350 1 ROUTINE tstres (depth) : NOVALUE =
226 0351 1
227 0352 1 ++
228 0353 1 FUNCTIONAL DESCRIPTION
229 0354 1
230 0355 1 This routine checks to see if SAVSTK is ok: i.e., certain open
231 0356 1 .SAVE/SAVE ALL and .RESTORE commands have been closed. If there are any
232 0357 1 unclosed commands it issues an error message and pops the stack.
233 0358 1
234 0359 1 FORMAL PARAMETERS:
235 0360 1
236 0361 1 DEPTH specifies the maximum allowed depth of .REQUIRE commands.
237 0362 1
238 0363 1 IMPLICIT INPUTS: None
239 0364 1
240 0365 1 IMPLICIT OUTPUTS: None
241 0366 1
242 0367 1 ROUTINE VALUE:
243 0368 1 COMPLETION CODES: None
244 0369 1
245 0370 1 SIDE EFFECTS: None
246 0371 1
247 0372 1 --
248 0373 1
249 0374 2 BEGIN
250 0375 2 OWN
251 0376 2 x;
252 0377 2
253 0378 2 LOCAL
254 0379 2 hold_iseqn,
255 0380 2 hold_page;
256 0381 2
257 0382 2 x = .savstk [0, savstk_depth];
258 0383 2
259 0384 2 IF (.x EQL 0) THEN
260 0385 2 RETURN; ! No unclosed .SAVEs.
261 0386 2
262 0387 2 IF .depth GTR .savstk [.x, savstk_req_d] THEN
263 0388 2 RETURN; ! Unclosed SAVEs but in files still open.
264 0389 2
265 0390 2 !There are unclosed saves to be reported.
266 0391 2
267 0392 2 erml (rnfms); ! Missing .ReStore commands, at end of file.
268 0393 2
269 0394 2 ! Now point the user back to the point where the corresponding
270 0395 2 ! SAVE/SAVEALL was made.
271 0396 2
272 0397 2 WHILE ( (.savstk [0, savstk_depth] GTR 0)
273 0398 2 AND
274 0399 2 (.depth LEQ .savstk [.savstk [0, savstk_depth], savstk_req_d]) ) DO
275 0400 2 BEGIN
276 0401 2 hold_page = .irac_ipagen;
277 0402 2 hold_iseqn = .irac_iseqn;
278 0403 2 irac_ipagen = .savstk [.savstk [0, savstk_depth], savstk_ipagen];
279 0404 2 irac_iseqn = .savstk [.savstk [0, savstk_depth], savstk_iseqn];
280 0405 2

```

PC	Instruction	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
----	-------------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

RSDERM
V04-000

Misc. error checks and messages.
TSTRES -- Check for, report, and restore missin

B 12
16-Sep-1984 01:40:15
14-Sep-1984 13:07:56

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]RSDERM.BLI;1

Page 11
(6)

FC	AS	53	DO	0008A	MOVL	HOLD_ISEQN, IRAC+8
	7E	01	CE	0008E	MNEGL	#1, =(SP)
	7E	8F	9A	00091	MOVZBL	#175, -(SP)
00000000G	EF	02	FB	00095	CALLS	#2, SAVRES
		9B	11	0009C	BRB	1\$
		04	0009E	3\$:	RET	

: 0412
: 0413
: 0397
: 0416

; Routine Size: 159 bytes, Routine Base: \$CODE\$ + 0087

RSK1
V04-

```

293 0417 1 GLOBAL ROUTINE tsttfe (depth) : NOVALUE =
294 0418 1
295 0419 1 ++
296 0420 1 FUNCTIONAL DESCRIPTION
297 0421 1
298 0422 1 This routine does nothing more than merely pass control to each of
299 0423 1 the respective test routines to check for unclosed lists, saves, etc.
300 0424 1 Oh, one exception: It now checks for being within a .LITERAL. If it
301 0425 1 finds this to be true, it issues an error and closes the literal.
302 0426 1
303 0427 1 FORMAL PARAMETERS:
304 0428 1
305 0429 1 DEPTH specifies the maximum allowed depth of .REQUIRE commands.
306 0430 1
307 0431 1 IMPLICIT INPUTS: None
308 0432 1
309 0433 1 IMPLICIT OUTPUTS: None
310 0434 1
311 0435 1 ROUTINE VALUE:
312 0436 1 COMPLETION CODES: None
313 0437 1
314 0438 1 SIDE EFFECTS: None
315 0439 1
316 0440 1 --
317 0441 1
318 0442 1 BEGIN
319 0443 1
320 0444 1 IF .gca_literal THEN ! If within a literal;
321 0445 1 BEGIN
322 0446 1 erm (rnflde ! say that it wasn't closed,
323 0447 1 ,CHSPTR(UPLIT('At or near end of document'))! and where
324 0448 1 ,26);
325 0449 1 lit( h_end_literal ) ! and then close it.
326 0450 1 END;
327 0451 1 tstblk (.depth); ! Check for unclosed lists, notes, and literals.
328 0452 1 tstcnd (.depth); ! Check for unclosed ifs and ifnots.
329 0453 1 tstres (.depth); ! Check for unclosed saves.
330 0454 1
331 0455 1 END; ! End of TSTTFE

```

```

20 64 6E 65 20 72 61 65 6E 20 72 6F 20 74 41 0000 P.AAA: .ASCII \At or near end of document\<0><0>
00 00 74 6E 65 6D 75 63 6F 64 20 66 6F 000F

```

```

.PSECT $CODES,NOWRT,2

.ENTRY TSTTFE, Save R2
MOVAB TSTBLK, R2
BLBC GCA+80, 1$
PUSHL #26
PUSHAB P.AAA
PUSHL #RNFLDE

0004 0000
52 FED4 CF 9E 00002
1E 00000000G EF E9 00007
00000000' 1A DD 0000E
00000000G 8F DD 00016

0417
0444
0446
0447
0446

```

: R

: S
: R
: E

RSDERM
V04-000

Misc. error checks and messages.

TSTRES -- Check for, report, and restore missin

D 12

16-Sep-1984 01:40:15

14-Sep-1984 13:07:56

VAX-11 BLISS-32 V4.0-742

[RUNOFF.SRC]RSDERM.BLI;1

Page 13

(7)

00000000G	EF		03	FB	0001C
			3C	DD	00023
00000000G	EF		01	FB	00025
		04	AC	DD	0002C
	62		01	FB	0002F
		04	AC	DD	00032
51	A2		01	FB	00035
		04	AC	DD	00039
0087	C2		01	FB	0003C
			04	00041	

1\$:

CALLS	#3	ERM
PUSHL	#60	
CALLS	#1	LIT
PUSHL	DEPTH	
CALLS	#1	TSTBLK
PUSHL	DEPTH	
CALLS	#1	TSTCND
PUSHL	DEPTH	
CALLS	#1	TSTRES
RET		

:	0449
:	
:	0451
:	
:	0452
:	
:	0453
:	
:	0455

; Routine Size: 66 bytes. Routine Base: \$CODE\$ + 0126

; 332 0456 1

RSK
V04

: L
: M
: C

```

334 0457 1 GLOBAL ROUTINE pntbac : NOVALUE =
335 0458 1 ++
336 0459 1 FUNCTIONAL DESCRIPTION
337 0460 1
338 0461 1 This routine saves the input file line/page information. It then
339 0462 1 substitutes information from FRMSTK for that information. Then ERM is
340 0463 1 called to output a message indicating where to look in the input file
341 0464 1 for the command that made the stack entry. But ERM does not have
342 0465 1 enough flexibility to accept all that information as parameters.
343 0466 1 However, it can pick it up from IRAC, which it does. In other words,
344 0467 1 this routine fakes out ERM a bit.
345 0468 1
346 0469 1 After ERM returns, the original information is restored.
347 0470 1
348 0471 1 --
349 0472 2 BEGIN
350 0473 2
351 0474 2 LOCAL
352 0475 2 hold_iseqn,
353 0476 2 hold_page;
354 0477 2
355 0478 2 hold_page = .irac_ipagen;
356 0479 2 hold_iseqn = .irac_iseqn;
357 0480 2 irac_ipagen = .frmstk [.frmstd, frmstk_ipagen];
358 0481 2 irac_iseqn = .frmstk [.frmstd, frmstk_iseqn];
359 0482 2 erm trnbak
360 0483 2 .frmstk [.frmstd, frmstk_fspecc]
361 0484 2 .frmstk [.frmstd, frmstk_fspecc] );
362 0485 2 irac_ipagen = .hold_page;
363 0486 2 irac_iseqn = .hold_iseqn;
364 0487 1 END;

```

!End of PNTBAC

```

003C 00000
55 00000000G FF 9E 00002
54 00000000G EF 9E 00009
52 64 D0 00010
53 FC A4 D0 00013
50 00000000G EF 0F C5 00017
64 6540 D0 0001F
FC A4 FC A540 D0 00023
08 A540 DD 00029
04 A540 DD 0002D
00000000G 8F DD 00031
00000000G EF 03 FB 00037
64 52 D0 0003E
FC A4 53 D0 00041
04 00045

```

```

.ENTRY PNTBAC, Save R2,R3,R4,R5
MOVAB FRMSTK-32, R5
MOVAB IRAC+12, R4
MOVL IRAC+12, HOLD_PAGE
MOVL IRAC+8, HOLD_ISEQN
MULL3 #15, FRMSTD, R0
MOVL FRMSTK-32[R0], IRAC+12
MOVL FRMSTK-36[R0], IRAC+8
PUSHL FRMSTK-24[R0]
PUSHL FRMSTK-28[R0]
PUSHL #RNFBAK
CALLS #3, ERM
MOVL HOLD_PAGE, IRAC+12
MOVL HOLD_ISEQN, IRAC+8
RET

```

```

0457
0478
0479
0480
0481
0484
0483
0482
0485
0486
0487

```

; Routine Size: 70 bytes, Routine Base: \$CODES + 0168

; 365 0488 1

367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397

0489 1 GLOBAL ROUTINE negind : NOVALUE =

0490 1

0491 1

0492 1

0493 1

0494 1

0495 1

0496 1

0497 1

0498 1

0499 1

0500 1

0501 1

0502 1

0503 1

0504 1

0505 1

0506 1

0507 1

0508 1

0509 1

0510 1

0511 1

0512 1

0513 2

0514 2

0515 2

0516 2

0517 2

0518 2

0519 1

++

FUNCTIONAL DESCRIPTION:

NEGIND keeps track of attempted indents past the left side of the page. To avoid issuing so many error messages that the user would be annoyed, it only issues an error message for the first such attempt. Other violations are simply counted; the summary count is output by REMNEG later.

FORMAL PARAMETERS: None

IMPLICIT INPUTS None

IMPLICIT OUTPUTS None

ROUTINE VALUE:

COMPLETION CODES: None

SIDE EFFECTS: None

--

BEGIN

IF .gca_nia EQL 0 THEN
erml(rnfnia);

gca_nia = .gca_nia + 1;

END;

!End of NEGIND

52	00000000G	EF	9E	00002	.ENTRY	NEGIND, Save R2	0489
		62	D5	00009	MOVAB	GCA+160, R2	0515
		0D	12	0000B	TSTL	GCA+160	0516
		8F	DD	0000D	BNEQ	1\$	0518
00000000G	EF	01	FB	00013	PUSHL	#RNFNIA	0519
		62	D6	0001A	CALLS	#1, ERML	
		04	0001C	1\$:	INCL	GCA+160	
					RET		

; Routine Size: 29 bytes, Routine Base: \$CODE\$ + 01AE

; 398 0520 1

```

400 0521 1 GLOBAL ROUTINE remneg : NOVALUE =
401 0522 1
402 0523 1 ++
403 0524 1 FUNCTIONAL DESCRIPTION:
404 0525 1
405 0526 1 Issues an error message indicating how many negative
406 0527 1 indents have been detected. Then resets the count to zero.
407 0528 1
408 0529 1 Used together with NEGIND.
409 0530 1
410 0531 1 FORMAL PARAMETERS: None
411 0532 1
412 0533 1 IMPLICIT INPUTS: None
413 0534 1
414 0535 1 IMPLICIT OUTPUTS: None
415 0536 1
416 0537 1 ROUTINE VALUE:
417 0538 1 COMPLETION CODES: None
418 0539 1
419 0540 1 SIDE EFFECTS: None
420 0541 1
421 0542 1 --
422 0543 1
423 0544 2 BEGIN
424 0545 2
425 0546 2 IF .GCA_NIA GTR 1 THEN ! 1 or zero means all accounted for.
426 0547 2 erm (rnfnic, .gca_nia - 1); ! Subtract off reported negative indent
427 0548 2
428 0549 2 gca_nia = 0; ! This starts the counting over again.
429 0550 1 END; ! End of REMNEG

```

			0004 0000	.ENTRY	REMNEG, Save R2	: 0521
	52 00000000G	EF 9E 00002		MOVAB	GCA+160, R2	: 0546
	01	62 D1 00009		CMPL	GCA+160, #1	: 0547
		11 15 0000C		BLEQ	1\$: 0549
7E	62	01 C3 0000E		SUBL3	#1, GCA+160, -(SP)	: 0550
	00000000G EF	8F DD 00012		PUSHL	#RNFNIC	
		02 FB 00018		CALLS	#2, ERMN	
		62 D4 0001F 1\$:		CLRL	GCA+160	
		04 00021		RET		

: Routine Size: 34 bytes, Routine Base: \$CODE\$ + 01CB

: 430 0551 1

```

: 432 0552 1 %SBTTL 'XMARG -- Controls R/L margin violation error messages'
: 433 0553 1 GLOBAL ROUTINE xmarg : NOVALUE =
: 434 0554 1
: 435 0555 1 ++
: 436 0556 1 FUNCTIONAL DESCRIPTION:
: 437 0557 1
: 438 0558 1 XMARG keeps track of attempted margin crossings (i.e., left
: 439 0559 1 margin exceeds right margin). To avoid issuing so many
: 440 0560 1 messages that the user would be annoyed, it only issues
: 441 0561 1 an error message for the first such attempt. Other violations
: 442 0562 1 are simply counted; the summary count is output by REMMRG later.
: 443 0563 1
: 444 0564 1 FORMAL PARAMETERS: None
: 445 0565 1
: 446 0566 1 IMPLICIT INPUTS: None
: 447 0567 1
: 448 0568 1 IMPLICIT OUTPUTS: None
: 449 0569 1
: 450 0570 1 ROUTINE VALUE:
: 451 0571 1 COMPLETION CODES: None
: 452 0572 1
: 453 0573 1 SIDE EFFECTS: None
: 454 0574 1
: 455 0575 1 --
: 456 0576 1
: 457 0577 2 BEGIN
: 458 0578 2
: 459 0579 2 IF .gca_xmarg EQL 0 THEN
: 460 0580 2 erml'(rnfmrgr);
: 461 0581 2
: 462 0582 2 gca_xmarg = .gca_xmarg + 1;
: 463 0583 1 END;

```

!End of XMARG

		0004 0000	.ENTRY XMARG, Save R2	: 0553
52	00000000G	EF 9E 00002	MOVAB GCA+164, R2	: 0579
		62 D5 00009	TSTL GCA+164	: 0580
		0D 12 0000B	BNEQ 1\$: 0582
	00000000G	8F DD 0000D	PUSHL #RNFMRG	: 0583
	EF	01 FB 00013	CALLS #1, ERML	
		62 D6 0001A 1\$:	INCL GCA+164	
		04 0001C	RET	

; Routine Size: 29 bytes, Routine Base: \$CODE\$ + 01ED

; 464 0584 1

```

: 466 0585 1 %SBTTL 'REMMRG -- reports # of crossed margin errors'
: 467 0586 1 GLOBAL ROUTINE REMMRG : NOVALUE =
: 468 0587 1
: 469 0588 1 ++
: 470 0589 1 FUNCTIONAL DESCRIPTION:
: 471 0590 1
: 472 0591 1 Issues an error message indicating how many crossed
: 473 0592 1 margins have been detected. Then resets the count to zero.
: 474 0593 1
: 475 0594 1 Used together with XMARG.
: 476 0595 1
: 477 0596 1 FORMAL PARAMETERS: None
: 478 0597 1
: 479 0598 1 IMPLICIT INPUTS: None
: 480 0599 1
: 481 0600 1 IMPLICIT OUTPUTS: None
: 482 0601 1
: 483 0602 1 ROUTINE VALUE:
: 484 0603 1 COMPLETION CODES: None
: 485 0604 1
: 486 0605 1 SIDE EFFECTS: None
: 487 0606 1
: 488 0607 1 --
: 489 0608 1
: 490 0609 2 BEGIN
: 491 0610 2
: 492 0611 2 IF .gca_xmarg GTR 1 THEN ! 1 or zero means all accounted for.
: 493 0612 2 erm (rnfmrc, .gca_xmarg - 1); ! Subtract off reported negative indent
: 494 0613 2
: 495 0614 2 gca_xmarg = 0; ! This starts the counting over again.
: 496 0615 1 END; ! End of REMMRG

```

			0004 00000	.ENTRY REMMRG, Save R2	: 0586
	52 00000000G	EF 9E 00002		MOVAB GCA+164, R2	: 0586
	01	62 D1 00009		CMPL GCA+164, #1	: 0611
		11 15 0000C		BLEQ 1\$: 0611
7E	62	01 C3 0000E		SUBL3 #1, GCA+164, -(SP)	: 0612
	00000000G EF	8F DD 00012		PUSHL #RNFMRC	: 0612
		02 FB 00018		CALLS #2, ERMN	: 0614
		62 D4 0001F 1\$:		CLRL GCA+164	: 0614
		04 00021		RET	: 0615

: Routine Size: 34 bytes, Routine Base: \$CODE\$ + 020A

```

: 497 0616 1
: 498 0617 1 END !End of module
: 499 0618 0 ELUDOM

```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	556	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$OWN\$	4	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$SPLIT\$	28	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	28	2	86	00:00.3

COMMAND QUALIFIERS

```

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RSDERM/OBJ=OBJ$:RSDERM MSRC$:RSDERM/UPDATE=(ENH$:RSDERM)
:
: Size:          556 code + 32 data bytes
: Run Time:      00:10.7
: Elapsed Time:  00:28.5
: Lines/CPU Min: 3478
: Lexemes/CPU-Min: 15303
: Memory Used:   58 pages
: Compilation Complete

```

0348 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RNODEF
LIS

RSDEM
LIS

RTERM
LIS

RUNOFF
LIS

RNOUMS
LIS

RSKIPS
LIS

SAURES
LIS

RNOURS
LIS

RNFEM
LIS